FIG. 1

STORAGE LAYER SERVER 46

EVENT DATA AND BLOOM FILTERS BY USER ID 64

DATABASE SOFTWARE (e.g., BERKELEY) 62

SYNCHRONIZATION SERVICE CODE 68

STORAGE LAYER SERVICE CODE 60

BLOOM FILTER GENERATION AND PROCESSING 78

CACHE LAYER SERVER 42

CACHE LAYER SERVICE CODE 70

UPDATE PROCESSING 72

QUERY PROCESSING 74

BLOOM FILTER GENERATION AND PROCESSING 76

CACHED EVENT DATA AND BLOOM FILTERS FOR RANGE OF SESSION IDs 43

FIG.2

File   Edit   View   Go   Favorite   Help

Back   Forw...   Stop   Refresh   Home   Search   Favorite   Print   Font   Mail

Address   WEBSEARCH.COM

YOUR ACCOUNT

WEB SEARCH
Results 1-10 of 2,850,000 for LORD OF THE RINGS:

PREVIOUSLY VIEWED ON FEB 14TH

(VIEWED)-LORDOFTHERINGSMOVIE.COM-LORD OF THE RINGS FAN PAGE
LORD OF THE RINGS CHRISTMAS GIFTS-DVD'S, GAMES, BOOKS, TOYS, & MORE!
COLLECTOR... THE LORD OF THE RINGS 10" DELUXE SAURON DOLL.<< CLICK...
DESCRIPTION: NEWS, POLLS AND BEHIND-THE-SCENES INFORMATION ON THE FILMS.
WWW.LORDOFTHERINGSMOVIE.COM/~7K-FEB 10, 2003

}100

NEW RESULT SINCE FEB.14TH

(NEW)-LORD OF THE RINGS-FANATICS SITE-FOR & BY TOLKIEN FANATICS
LORD OF THE RINGS SITE FOR AND BY TOLKIEN FANATICS: GUIDED TOURS OF MIDDLE-EARTH,
111-QUESTION-QUIZ, ART COLLECTIONS, CREATIVE AND PRODUCT SPECIALS, SOUND CLIPS...
DESCRIPTION: GUIDED TOURS OF MIDDLE-EARTH, 111-QUESTION QUIZ, AND SPECIALS.
WWW.LORDOTRINGS.COM/~15K-FEB 10, 2003-

}120

TOLKIENMOVIES.COM-LORD OF THE RINGS MOVIE NEWS, RUMORS, PHOTOS...
YOUR SOURCE FOR EVERYTHING CONCERNING THE UPCOMING LORD OF THE RINGS MOVIES BY
NEW LINE CINEMA...VIEW MORE DEBATE...THE LORD OF THE RINGS -THE TWO TOWERS...
DESCRIPTION: NEWS, CASTING INFORMATION, DISCUSSION, AND RUMORS.
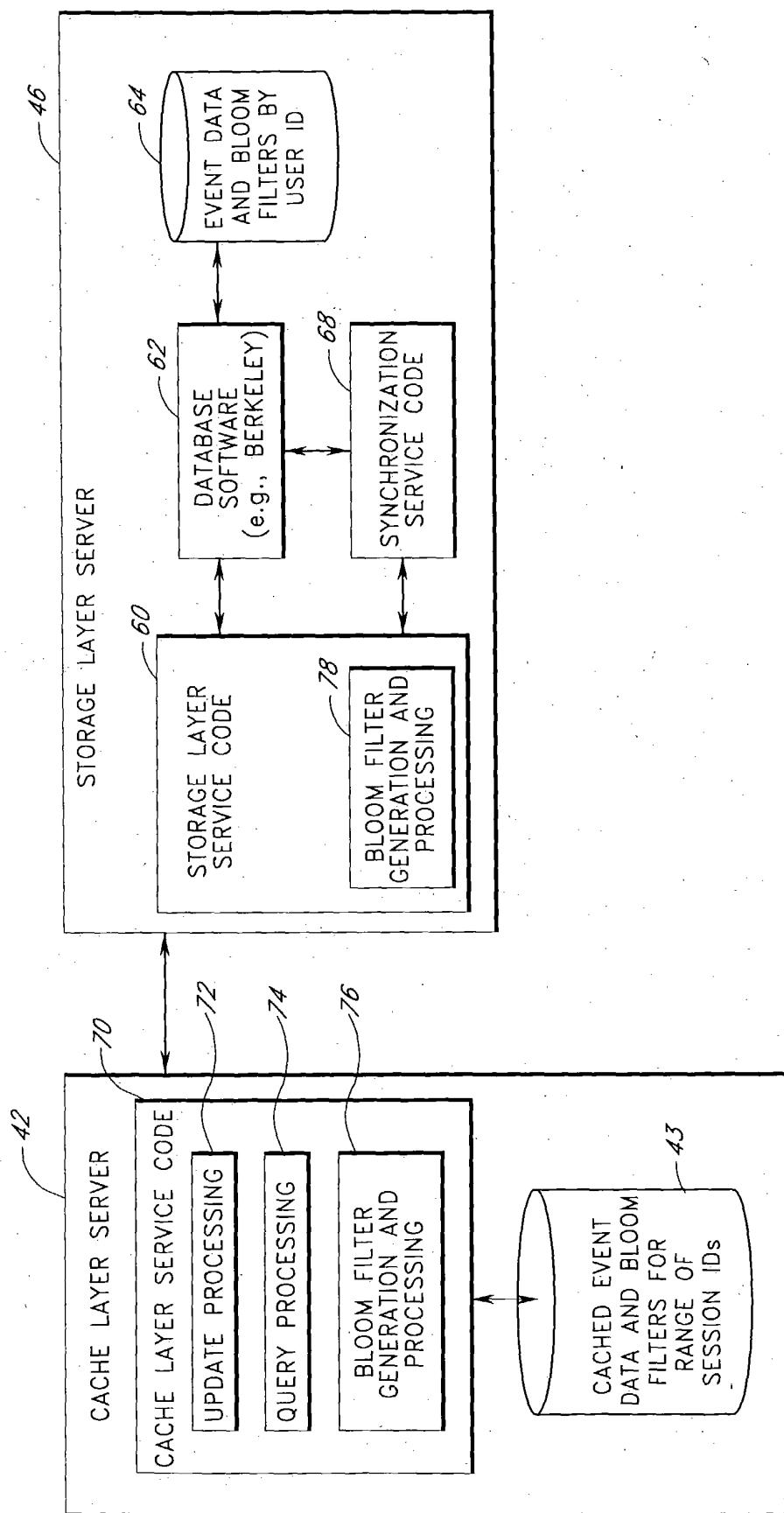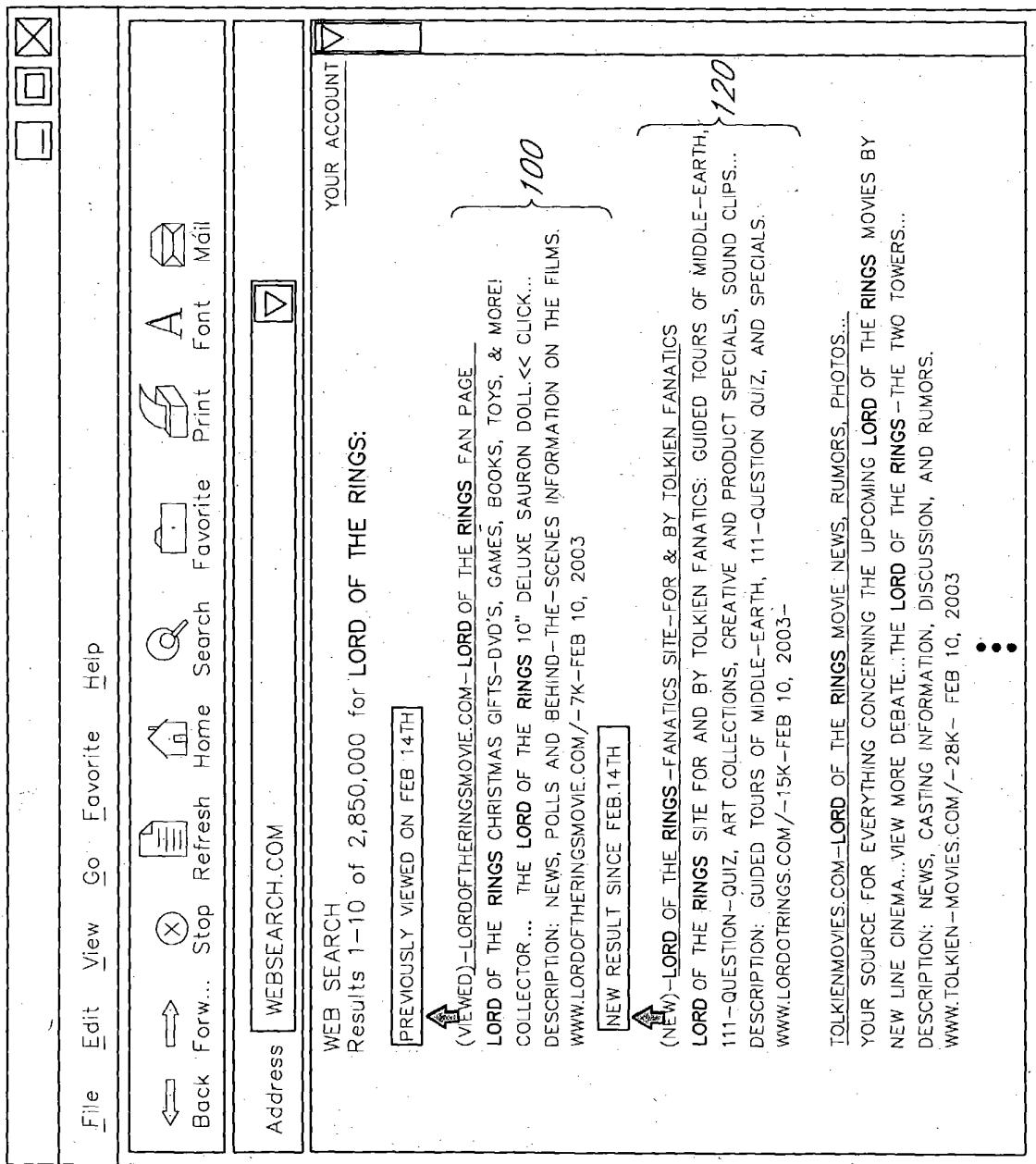WWW.TOLKIEN-MOVIES.COM/~28K- FEB 10, 2003

• • •

*FIG.3*

SERVER ARCHITECTURE AND METHODS FOR PERSISTENTLY STORING AND SERVING EVENT DATA
Van Vleet, et al.
Appl. No.: UNKNOWN   Atty Docket: ZNET.093A

4/7

```
┌────────────────────────────────┐
│    RECEIVE  SEARCH  QUERY       │⟋ 110
│     AND  EXECUTE  SEARCH        │
└────────────────────────────────┘
                │
                ▼
┌────────────────────────────────┐
│   FOR  EACH  SEARCH  RESULT  ITEM │
│  (EXTERNAL  URL,  CATALOG  ITEM, ETC.), │⟋ 112
│     QUERY  EVENT  HISTORY  SERVER │
│   TO  DETERMINE  WHETHER  USER  HAS │
│          ACCESSED  ITEM         │
└────────────────────────────────┘
                │
                ▼
┌────────────────────────────────┐
│     FOR  EACH  ACCESSED  ITEM,  │
│      QUERY  EVENT  HISTORY      │⟋ 114
│       SERVER  FOR  DATE  OF     │
│          LAST  ACCESS           │
└────────────────────────────────┘
                │
                ▼
┌────────────────────────────────┐
│   GENERATE  SEARCH  RESULTS  PAGE │⟋ 116
│  WITH  VIEWED-ITEM  ANNOTATIONS │
└────────────────────────────────┘
```
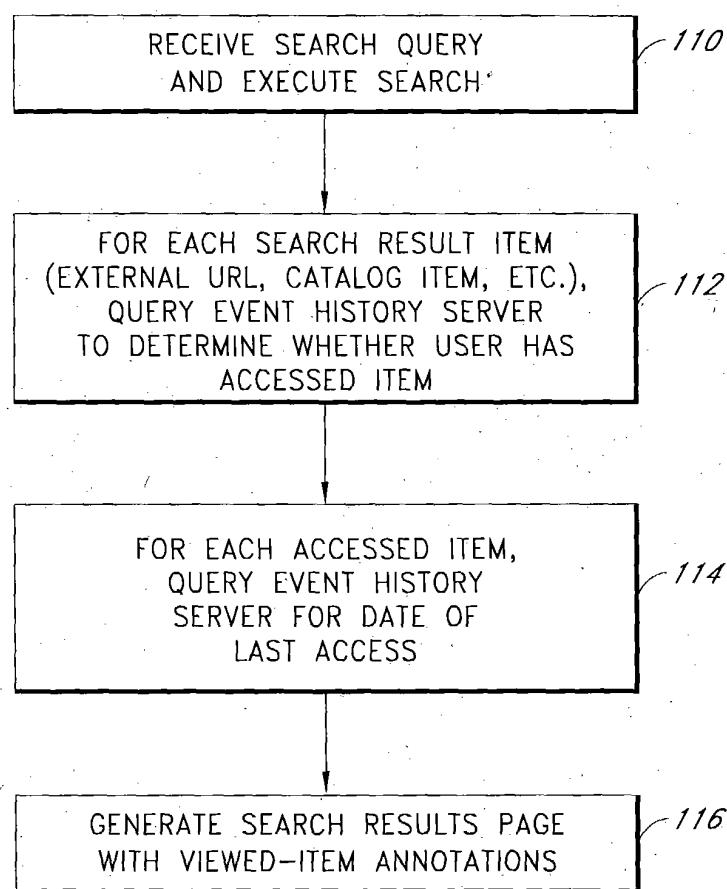
# FIG.4

*SERVER ARCHITECTURE AND METHODS FOR PERSISTENTLY STORING AND SERVING EVENT DATA*
*Van Vleet, et al.*
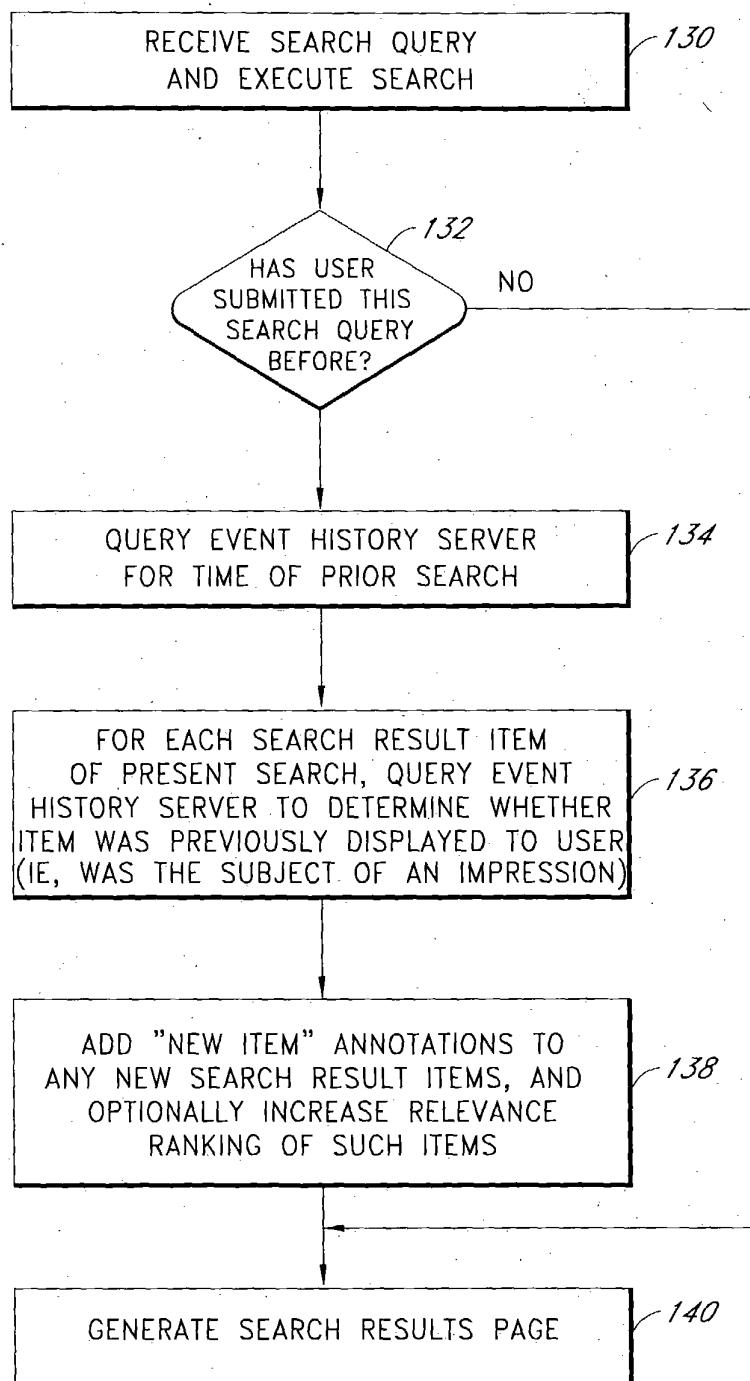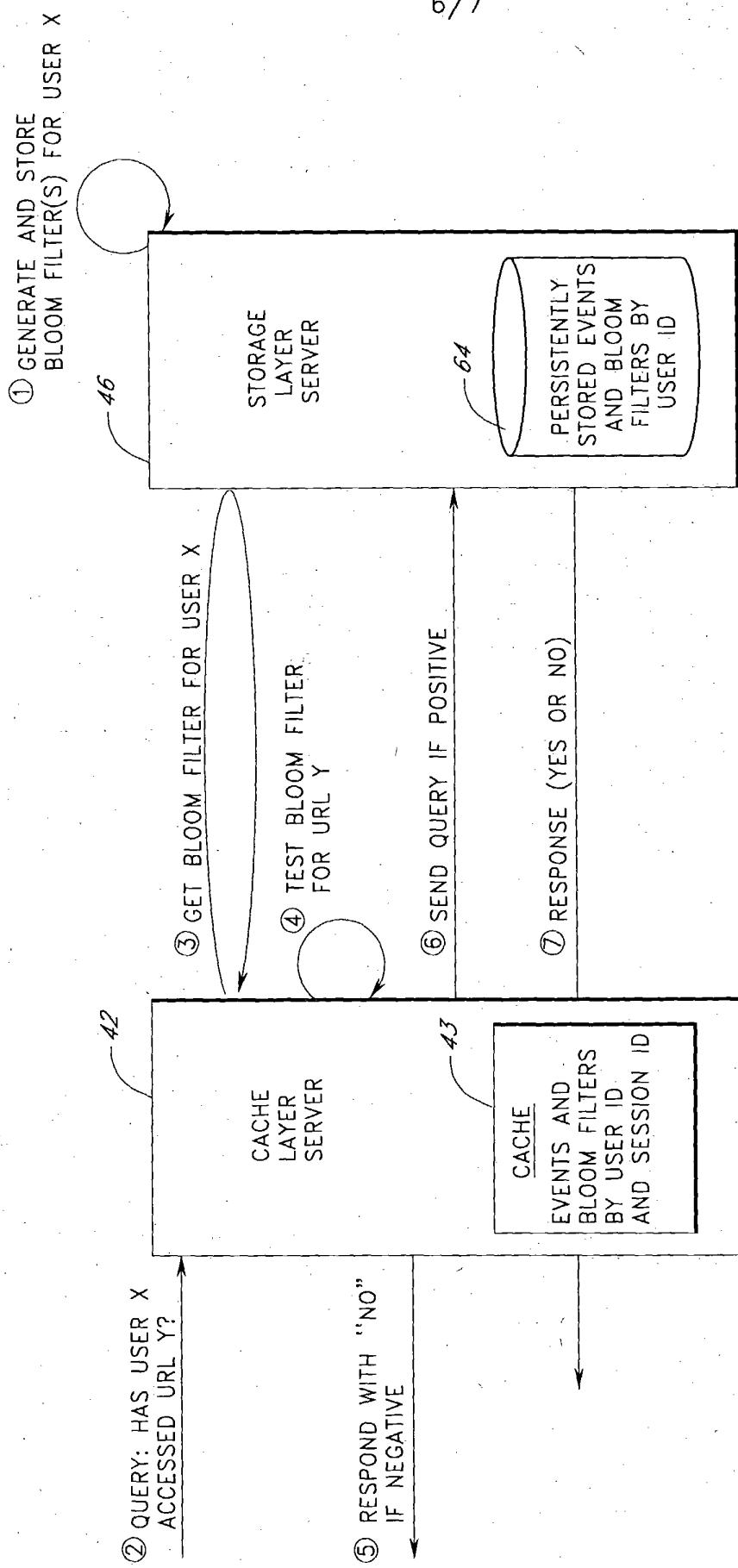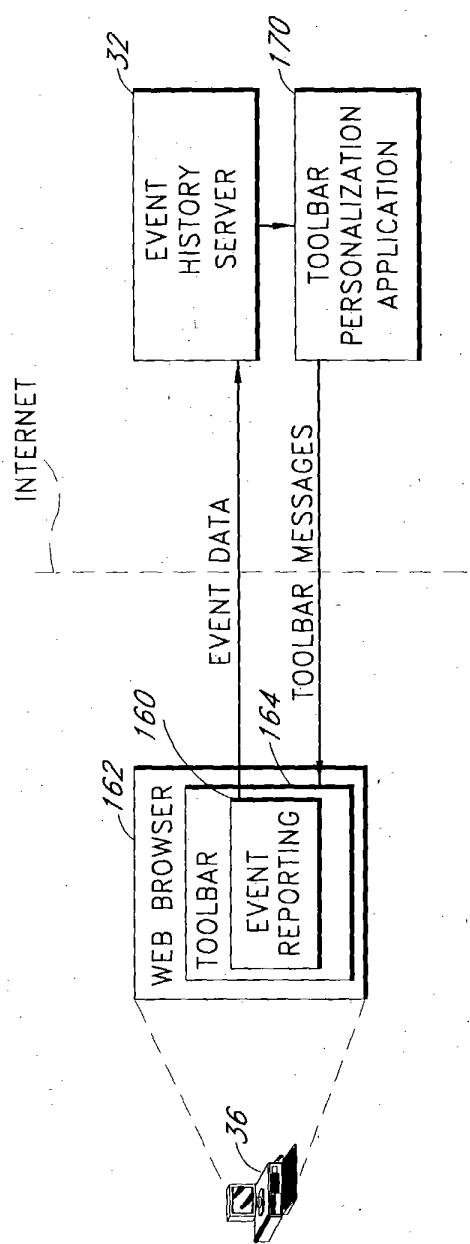*Appl. No.: UNKNOWN   Atty Docket: ZNET.093A*

5/7

```
          ┌─────────────────────────────┐
          │    RECEIVE  SEARCH  QUERY    │ ⟍ 130
          │    AND  EXECUTE  SEARCH      │
          └─────────────────────────────┘
                         │
                         ▼
                      ╱────╲  ⟋ 132
                   ╱ HAS USER ╲
                 ╱ SUBMITTED THIS ╲      NO
                ⟨  SEARCH QUERY    ⟩─────────────┐
                 ╲   BEFORE?     ╱                │
                   ╲           ╱                  │
                      ╲────╱                      │
                         │                        │
                         ▼                        │
          ┌─────────────────────────────┐         │
          │  QUERY  EVENT  HISTORY  SERVER │ ⟍ 134  │
          │    FOR  TIME  OF  PRIOR  SEARCH │       │
          └─────────────────────────────┘         │
                         │                        │
                         ▼                        │
          ┌─────────────────────────────┐         │
          │    FOR  EACH  SEARCH  RESULT  ITEM │   │
          │  OF  PRESENT  SEARCH,  QUERY  EVENT │ ⟍ 136
          │ HISTORY SERVER TO DETERMINE WHETHER │ │
          │ITEM WAS PREVIOUSLY DISPLAYED TO USER│ │
          │(IE, WAS THE SUBJECT OF AN IMPRESSION)│ │
          └─────────────────────────────┘         │
                         │                        │
                         ▼                        │
          ┌─────────────────────────────┐         │
          │   ADD  "NEW  ITEM"  ANNOTATIONS  TO │ │
          │ ANY NEW SEARCH RESULT ITEMS, AND │ ⟍ 138
          │  OPTIONALLY INCREASE RELEVANCE │     │
          │    RANKING  OF  SUCH  ITEMS     │     │
          └─────────────────────────────┘         │
                         │◄───────────────────────┘
                         ▼
          ┌─────────────────────────────┐
          │  GENERATE  SEARCH  RESULTS  PAGE │ ⟍ 140
          └─────────────────────────────┘
```

# FIG.5

① GENERATE AND STORE BLOOM FILTER(S) FOR USER X

46

STORAGE LAYER SERVER

64

PERSISTENTLY STORED EVENTS AND BLOOM FILTERS BY USER ID

③ GET BLOOM FILTER FOR USER X

④ TEST BLOOM FILTER FOR URL Y

⑥ SEND QUERY IF POSITIVE

⑦ RESPONSE (YES OR NO)

42

CACHE LAYER SERVER

43

CACHE

EVENTS AND BLOOM FILTERS BY USER ID AND SESSION ID

② QUERY: HAS USER X ACCESSED URL Y?

⑤ RESPOND WITH "NO" IF NEGATIVE

FIG.6

FIG. 7